Second International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, September 29 - October 02, 2011

On an approach for cubic Bézier interpolation

Dana Simian, Corina Simian

Abstract

The aim of this paper is to introduce an interpolation approach using cubic Bézier curves and surfaces which shape is controlled by means of two parameters. Implementation of the proposed method is realized using MATLAB.

1 Introduction

Bézier curves and surfaces have a wide applicability in Computer Aided Geometric Design (CAGD). In the last years, many studies were dedicated to obtain new classes of Bézier curves and surfaces suitable for the approximation process of various shapes. One direction of generalizations consists of replacing the Bernstein basis in the Bézier curves and surfaces parametric expression with a generalized Bernstein basis. Bernstein–Stancu polynomials, q-Bernstein polynomials, q-Bernstein–Stancu polynomials are examples of this kind of bases ([4], [6], [7], [8], [12]). The shape preserving properties of various operators which generalize the Bernstein operator are essential in defining new generalization of Bernstein basis which can be used in Bézier curves and surfaces theory. The quasi-Bézier curves and surfaces are based on a class of polynomial basis functions with n adjustable shape parameters ([5]). In [11], umbral calculus is used to generalize Bernstein polynomials and Bézier curves. Farin, introduced in [3] a class of 3D A- Bézier curves, defined by their degree, a vector v and a matrix M. These curves are used in the design of those parts of a car which are critical for its aesthetic appearance: parts of the hood, fender, or roof. An embedded Bézier shape parameterizations is constructed and employed in [1], to define multi-level optimum-shape algorithms.
In practice, a designer needs to fit a curve to digitized points. The aim of this paper is to study and implement an approach to design an interpolation cubic Bézier curve which passes through 4 digitizes points and depends on two parameters which control the curve's shape. Using tensor product method we define also an interpolation cubic Bézier surface.
The article is organized as follow: in the section 2 we present the parametric and matrix representation of Bézier curves and surfaces. In section 3 is introduces the proposed approach for obtaining interpolation cubic Bézier curves and surfaces. Section 4 is dedicated to the implementation and results' analysis. Conclusion and further directions of study can be found in section 5.

2 Bézier curves and surfaces

Bézier curves and surfaces are parametric curves and surfaces expressed in Bernstein basis using a set of control points as coefficients. The most used are the cubic Bézier curves and surfaces and we will referee to them in the following of the paper.
The equation of a cubic Bézier curves is:

$$c(t) = (x(t), y(t), z(t)) = \sum_{i=0}^{3} b_i \cdot B_i^3(t), t \in [0,1], \tag{1}$$

$$B_i^3(t) = \begin{pmatrix} i \\ 3 \end{pmatrix} t^i (1-t)^{3-i} \tag{2}$$

being the Bernstein polynomials of degree 3 and $b_i = (bx_i, by_i, bz_i)$ the control points which form the control polygon.
Remark 1. *A cubic Bézier curve is determined by its 4 control points.*

The properties that make the Bézier curves suitable for CAGD applications are affine invariance, (invariance under affine transformations), convex hull property (Bézier curve lies in the convex hull of the control points), endpoint interpolation and pseudo – local control (a change of one of the control point affects the Bézier curve only in the region of this control point). More details about Bézier curves properties can be found in the Farin comprehensive book [2].
A cubic Bézier surface is given by the equation:

$$s(u,v) = \sum_{i=0}^{3} \sum_{j=0}^{3} b_{ij} \cdot B_j^3(u) \cdot B_i^3(v), u, v \in [0,1] \ , \tag{3}$$

$b_{ij} = (bx_{ij}, by_{ij}, bz_{ij})$ being the control points of the surface and $B_j^3(u), B_i^3(v)$ the Bernstein polynomials given in (2).

Remark 2. *A Bézier surface is determined by its 16 control points*.

The properties enumerated for Bézier curves are also valid for Bézier surfaces.
For computational reasons it is very useful to transform the equations (1) and (3) in matrix form.
Matrix form of a Bézier curve:

$$c(t) = b \cdot B(t), \text{ with } c(t) \in \mathrm{M}_{3,1}, b \in \mathrm{M}_{3,4}, B(t) \in \mathrm{M}_{4,1}, t \in [0,1] \tag{4}$$

c(t) represents the column vector of the coordinates of a point from the Bézier curve corresponding to the value *t* of the parameter.
b is the matrix of control points coordinates. Each column correspond to a control point coordinates.
B(t) is the column vector of Bernstein polynomials computed for the value *t* of the parameter.
Matrix form of a Bézier surface:

$$s(u,v) = B'(v) \cdot b \cdot B(u) \tag{5}$$

(5) represents in fact 3 equations, one for each coordinate

$$x(u,v) = B'(v) \cdot bx \cdot B(u)$$

$$y(u,v) = B'(v) \cdot by \cdot B(u)$$

$$z(u,v) = B'(v) \cdot bz \cdot B(u)$$

$x(u,v), y(u,v), z(u,v) \in \mathrm{M}_{1,1} \equiv \mathbf{R}$ represent the coordinates of a point from the Bézier surface, corresponding to the values *u* and *v* of the parameters.
$bx, by, bz \in \mathrm{M}_{4,4}$ are matrixes containing the coordinates of the 16 control points of the surface.
B(u), B(v) are the column vectors of Bernstein polynomials computed for the values *u* and *s*.
We denoted by *M'* the transpose of the matrix *M.*

3. Bézier interpolation. Main results.

The approximation problem of more complicated curves has been solved by using piecewise Bézier curves satisfying different conditions in the junction points. From a practical point of view piecewise cubic Bézier curves of class C^1 and G^1 are preferred. Many interpolation problems as well as many solutions have been formulated: C^1piecewise cubic Hermite interpolation, point-normal interpolation, F-Mill interpolation (see [2]), etc. The most complex design problems of curves and surfaces require techniques related to Bézier splines, B-splines and NURBS. A brief presentation of these concepts can be found in [10] and many mathematical and algorithmic details are included in [2].
Our aim is to present and to implement a method for solving an interpolation problem using a single Bézier curve dependent of two parameters and to generalize the method in the case of surface interpolation using tensor product method. We will show that the Bézier curve shape is strong influenced by the two parameters.

Problem 1. (Interpolation using Bézier curve)
Being given 4 points on a curve C, find a cubic Bézier curve that passes through the given points. We suppose that two of the given points are the endpoints of the curve C.

Let $P_i, i \in \{0,...,3\}$ be the interpolation points. They satisfy the equation of the Bézier curve for some values of the parameter *t*.
Taking into account the *Remark 1*, for solving the *Problem 1* it is sufficient to find the 4 control points.
From the endpoint interpolation property of the Bézier curve, we have obviously

$$c(0) = P_0 = b_0 \tag{6}$$

and

$$c(1) = P_3 = b_3 \tag{7}$$

Let $t_1 < t_2 \in (0,1)$ be the values of parameters for which $c(t_i)=P_i,\ i \in \{1,2\}$. Then

$$c(t_i) = (1-t_i)^3 b_0 + 3t_i(1-t_i)^2 b_1 + 3t_i^2(1-t_i) + t_i^3 b_3,\ i \in \{1,2\} \tag{8}$$

The system (6), (7), (8) can be formulated as

$$A \cdot b = P \tag{9}$$

with

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ (1-t_1)^3 & 3t_1(1-t_1)^2 & 3t_1^2(1-t_1) & t_1^3 \\ (1-t_2)^3 & 3t_2(1-t_2)^2 & 3t_2^2(1-t_2) & t_2^3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$

and

$$b = \begin{pmatrix} bx_0 & by_0 & bz_0 \\ bx_1 & by_1 & bz_1 \\ bx_2 & by_2 & bz_2 \\ bx_3 & by_3 & bz_3 \end{pmatrix};\ P = \begin{pmatrix} Px_0 & Py_0 & Pz_0 \\ Px_1 & Py_1 & Pz_1 \\ Px_2 & Py_2 & Pz_2 \\ Px_3 & Py_3 & Pz_3 \end{pmatrix}$$

We have

$$\det(A) = 9t_1t_2(1-t_1)(1-t_2)(t_2-t_1) \neq 0 \tag{11}$$

The control points are given by

$$b = A^{-1} \cdot P \tag{12}$$

From (11) we have that

$$\lim_{t_i \to 0} A(t_1,t_2) = \lim_{t_i \to 1} A(t_1,t_2) = \lim_{t_1 \to t_2} A(t_1,t_2) = 0 \tag{13}$$

In practice, the points P_i are usually obtained by measurements. A method which considers that the points are uniformly distributed and correspond to the values of parameters $t_1 = \frac{1}{3}, t_2 = \frac{2}{3}$ on the interpolation curve has been used before, but it is difficult to estimate exactly these points and for the complicated shape of the curve the method does not offer good results. The values t_1 and t_2 from our approach can be modified in order to obtain an enough complicated shape by using a single cubic Bézier curve. More, it is not necessary an expensive method to measure the coordinates of the desired interpolation points.
The conclusion is that it is of interest to implement relation (12) such that using a continuous variation of the two parameters to obtained the desired shape.
Relation (13) suggests us that exist real values ε, δ and λ such that for

$$t_1 < \varepsilon, 1 - t_2 > \delta \text{ and } |t_1 - t_2| < \lambda \tag{14}$$

the matrix A from (12) is close to a singular matrix. The parameters t_i, $i \in \{1,2\}$, satisfying the inequalities (14) with the corresponding values ε, δ and λ small enough will be call *limit parameters* and we will refer to the inequalities from (14) as *limit inequalities*.
So it is interesting to study the dependence of the shape properties depending on the relative position of these two parameters.
The implementation details will be presented in section 4.

Problem 2. (Interpolation using Bézier curve)
Being given 16 points on a surface S, find a cubic Bézier surface that passes through the given points.
Let be $P_i, i \in \{0,...,15\}$ the interpolation points. The surface is generated using the tensor- product method. Let denote by $px, py, pz \in \mathrm{M}_{4,4}$ the matrixes containing the coordinate of the 16 interpolation points. In the tensor- product method we apply relation (12) first for the lines of the matrix px, py, pz and then for the columns of the new obtained matrix. The values t1, t2 were chosen the same for all the lines of the three matrixes. Another possibility is to choose different values t_1^j, t_2^j for the lines $j \in \{1,2,3\}$. The observation regarding the limits values for the parameters t_1, t_2 remain the same as in the case of Bézier curve.

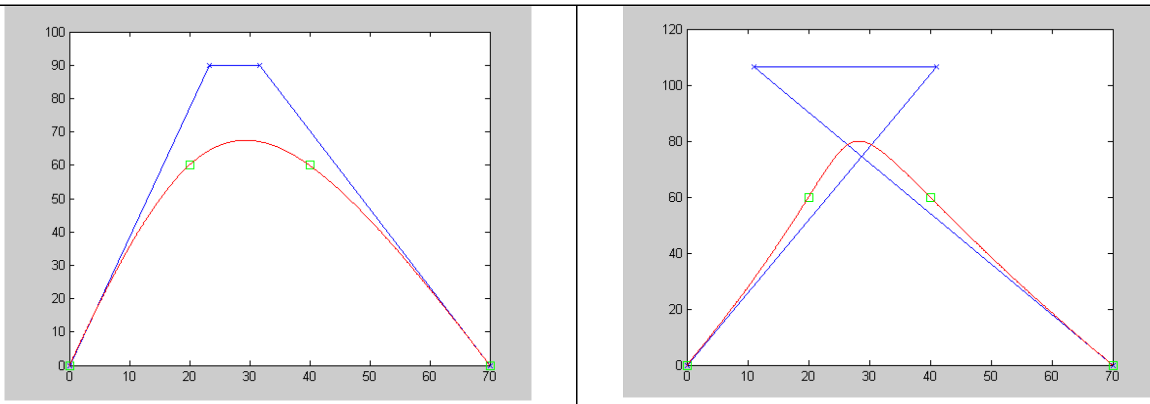
4 Implementation details and results

We implemented our proposed approach, for curves and surfaces interpolation, using MATLAB. The main reasons for which we choose MATLAB were:
- Capabilities to easy manipulate and operate with matrix structures
- Powerful function for graphical representation
- Possibility of integration of programs in a high quality Graphical User Interfaces

In [9] we presented a software system implemented in MATLAB for the shape design of punches used in deformation process and the analysis and comparison of the behaviour of different materials function of the desired shapes.

4.1 Implementation of our approach for interpolation of curves.

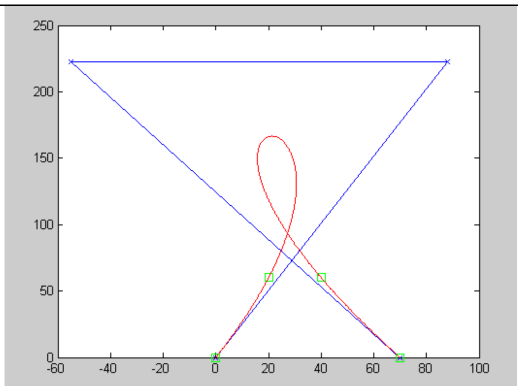
The computation of the Bézier curve points is made using the equation (4). The implemented function returns the coordinates of the control points. For the input data $P = \begin{pmatrix} 0 & 20 & 40 & 70 \\ 0 & 60 & 60 & 0 \end{pmatrix}$, the influence of the parameters t_1, t_2 is illustrated in the figures below:



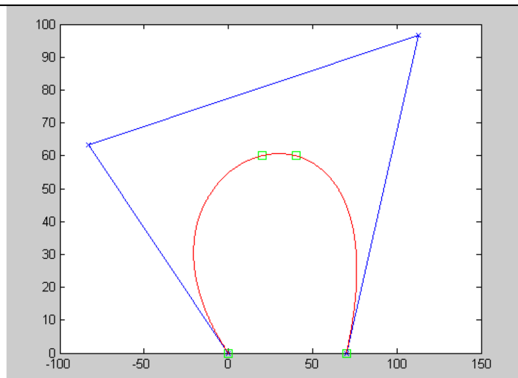
t1=1/3, t2=2/3 t1=1/4, t2=3/4

Fig.1 - Bézier curves obtained for 0<t1<0.5<t2<1

The convexity of the two curves is different. The equidistant points lead to a curve without inflexion points.



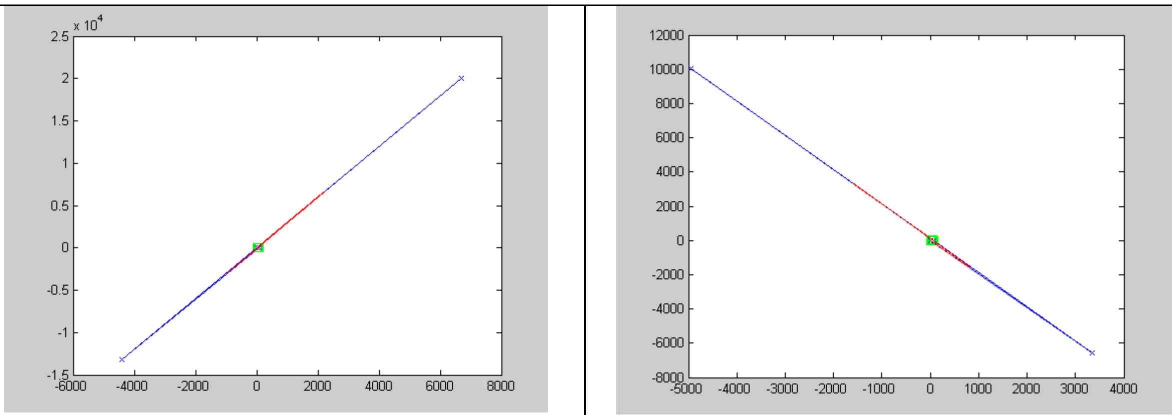
t1=0.1, t2=0.9



t1=0.5, t2=0.6

Fig. 2a - Bézier curve with t1=0.1, 1-t2=0.1 Fig. 2b - Bézier curve with t2-t1=0.1

In Fig. 2a and Fig. 2b the parameters t_i satisfy the limit inequalities (13) with the values of ε, δ and λ equal to 0.1.



t1=0.001, t2=0.6 t1=0.4, t2=0.999

Fig. 3 – Bézier curves obtained for limit parameters ε =0.001, δ=0.001

For a small value of the limit parameters ε (left graphic) or δ (right grahic), the Bézier curves degenerate in a straight line.

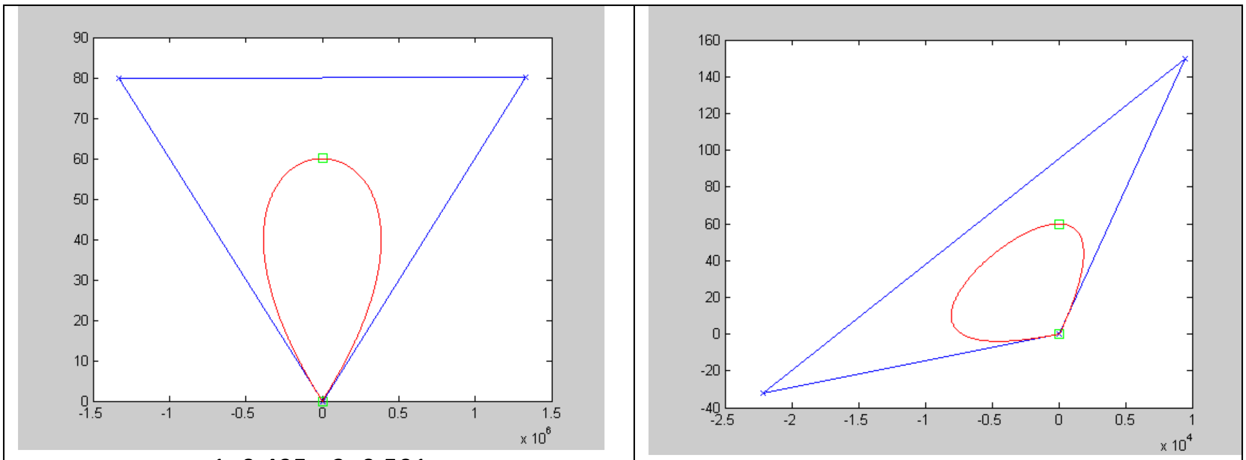

t1=0.495, t2=0.501 t1=0.7, t2=0.701

Fig. 4 – Bézier curves obtained for limit parameter λ =0.001

For a small value of limit parameter λ, the control polygon reduces to a triangle i.e. the cubic Bézier curve reduces to a quadratic one.

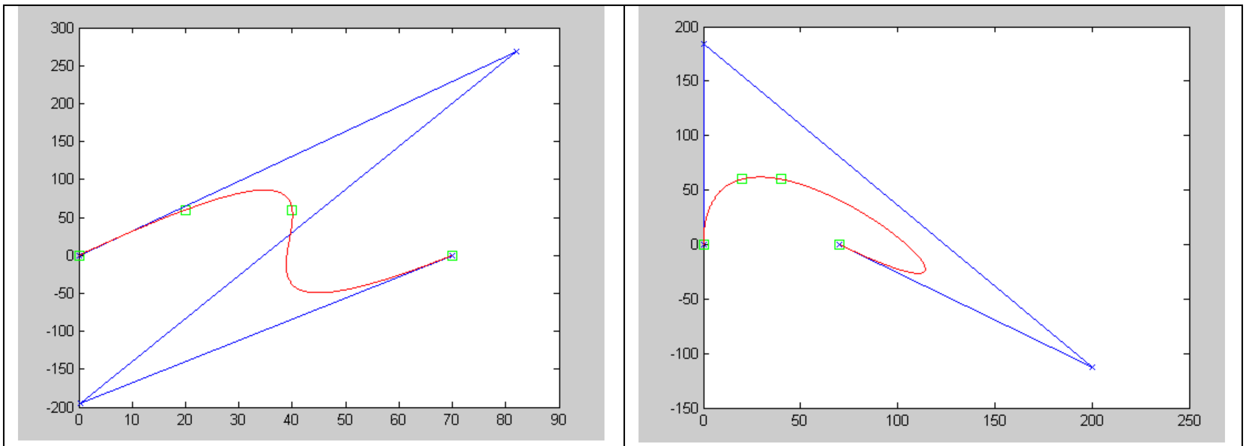

t1=0.1, t2=0.3 t1=0.2, t2=0.3

Fig. 5 - Bézier curves obtained for 0<t1<t2<0.5

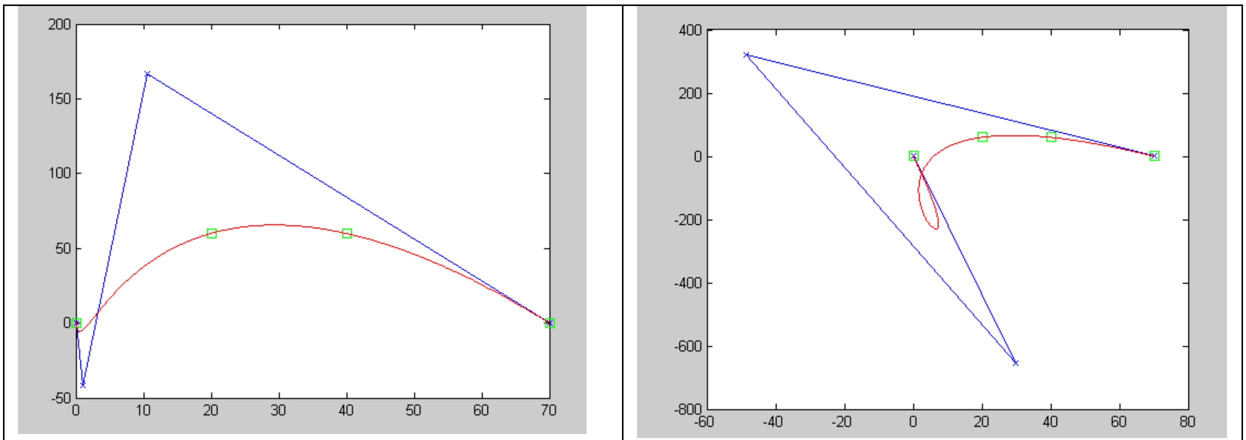
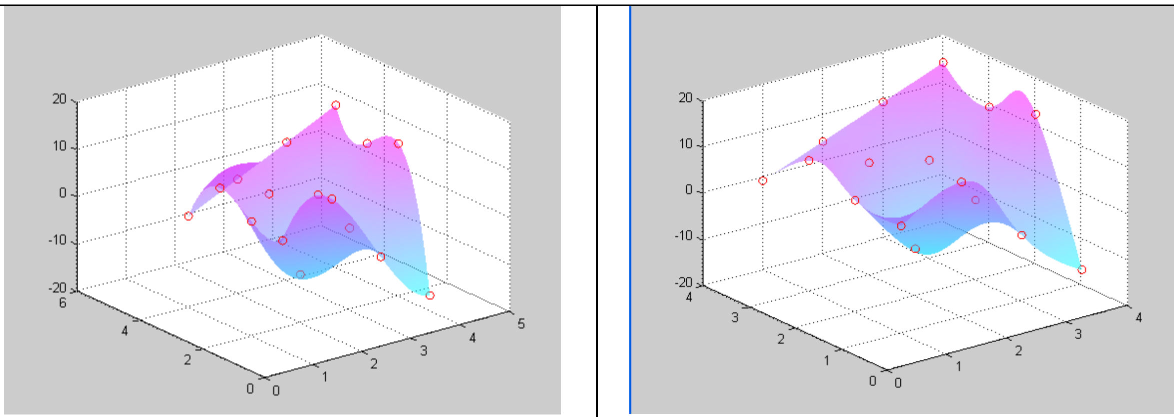

t1=0.6, t2=0.8 t1=0.8, t2=0.9

Fig. 6 - Bézier curves obtained for 0.5<t1<t2<1

If the parameters are located in the same half of the interval (0, 1), more complicated shapes are obtained.

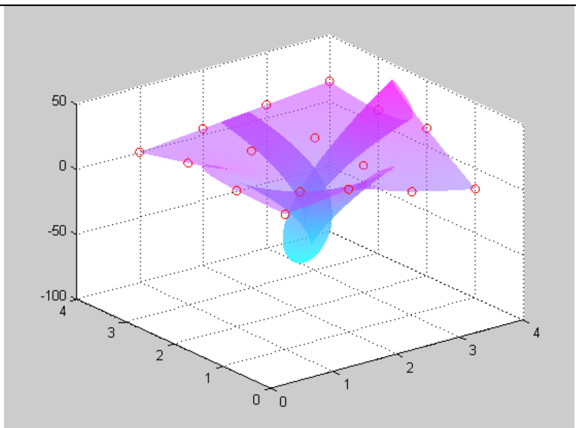
4.2 Implementation of our approach for interpolation of surfaces.

In the case of surfaces interpolation we implemented the tensor product method for the coordinates x, y and z, using the interpolation function defined for curve interpolation.



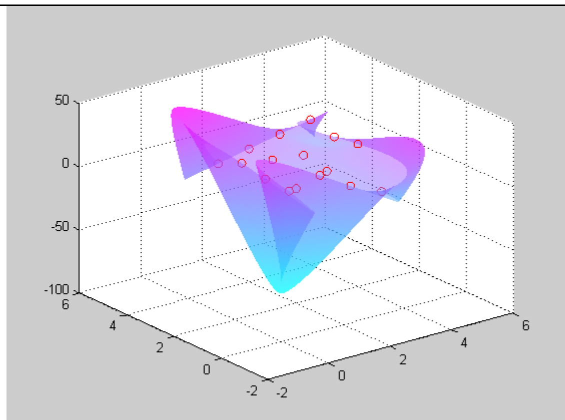
. t1=1/3, t2=2/3 t1=1/4, t2=3/4

Fig.7 - Bézier surfaces obtained for 0<t1<0.5<t2<1



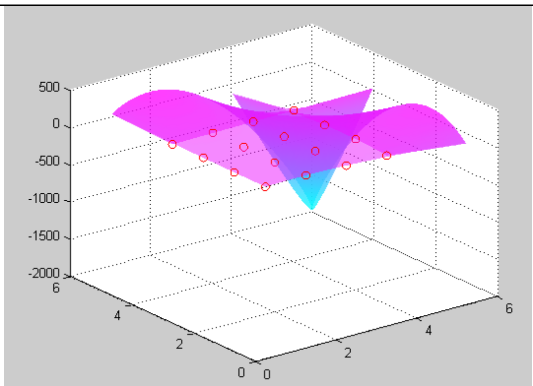
t1=0.1, t2=0.9

Fig. 8a - Bézier surface for t1=0.1, 1-t2=0.1



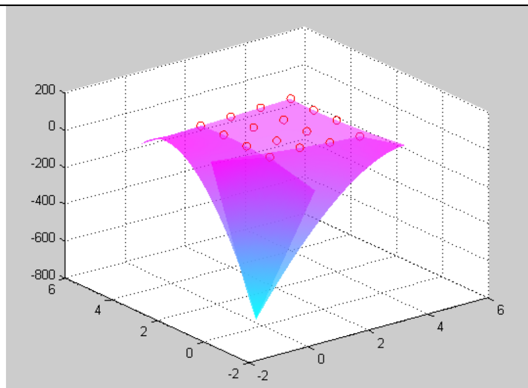
t1=0.5, t2=0.6

Fig. 8b - Bézier surface for t2-t1=0.1



t1=0.1, t2=0.2

Fig. 9a - Bézier surface for 0<t1<t2<0.5



t1=0.8, t2=0.9

Fig. 9b - Bézier surface for 0.5<t1<t2<1

The surfaces presented in the Fig. 7 – Fig. 9 were obtained for the input data

$$Px = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix},\ Py = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix},\ Py = \begin{pmatrix} 3 & 9 & -6 & -17 \\ 4 & -10 & -3 & 12 \\ 8 & 4 & 1 & 9 \\ -1 & 4 & 9 & 14 \end{pmatrix}$$

5 Conclusions and further directions of study

In this paper we propose an approach which allows us to modify in an easy way the shape of a cubic Bézier interpolation curve and surface using two real parameters. The relative position of these parameters and their distance from 0 (for the first parameter) and respective 1 (for the second parameter) has a significant influence. An advantage of the method is that the interpolation points can be situated anywhere on the original curve or surface and the convexity properties of the interpolation curve or surface can be obtained by parameters modification. We can also use our cubic Bézier interpolation curves in order to obtain a piecewise Bézier curve of G^1 class.
A further direction of study is represented by the mathematical foundation of the limit cases of the parameters and mathematical study of the parameters' values –shape dependence.

References

[1] J.-A. Désidéri, *Hierarchical Optimum-Shape Algorithms, Using Embedded Bezier Parameterizations,* Numerical Methods for Scientific Computing Variational Problems and Applications, Y. Kuznetsov, P. Neittanmaki and O. Pironneau (Eds.) CIMNE, Barcelona, 1-12, 2003.

[2] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design,* Academic Press, Boston, 1996.

[3] G. Farin, *Class A Bezier Curves,* Computer Aided Geometric Design 23, Elsevier Press, 573-581, 2006.

[4] A.D. Gadjiev, A.M. Ghorbanalizadeh, *Approximation properties of new type Bernstein–Stancu polynomials of one and two variables,* Applied Mathematics and Computation 216, 890–901, 2010.

[5] Xi-An Han, YiChen Ma, XiLi Huang, *A novel generalization of Bézier curve and surface, Journal of Computational and Applied Mathematics, 217, 180-193, 2008.*

[6] G. Nowak, *A de Casteljau Algorithm for q-Bernstein-Stancu Polynomials,* Hindawi Publishing Corporation Abstract and Applied Analysis, Volume 2011, Article ID 609431, 13 pages doi:10.1155/2011/609431.

[7] Halil Oru, George M. Phillips, *q-Bernstein polynomials and Bézier curves,* Journal of Computational and Applied Mathematics*, 15, 1-12,* 2003.

[8] G. M. Phillips, *A de Casteljau Algorithm for Generalized Bernstein Polynomials,* BIT 36:1, 232-236, 1996.

[9] D. Simian, C. Simian, *Bézier Techniques Applied in a Complex Virtual Testing System Design,* Academic Journal of Academic Engineering, Vol. 8, Issue 2/2010, 75-81, 2010.

[10] D. E. Ulmet, *Swept Surfaces in Computer Aided Geometric Design,* Proceedings of International workshop New Approaches, Algorithms and Advanced Computational Techniques in Approximation Theory and its Applications, "Lucian Blaga" University Press, Sibiu, 19-31, 2007.

[11] R. Winkel, *Generalized Bernstein Polynomials and Bézier Curves: An Application of Umbral Calculus to Computer Aided Geometric Design, 1-25, 2001.*

[12] Lianying Yun, Xueyan Xiang, *On Shape-Preserving Properties and Simultaneous Approximation of Stancu Operator,* Analysis in Theory and Applications, Volume 24, Number 2, 195–204, 2008, DOI:10.1007/s10496-008-0195-0.

DANA SIMIAN
"Lucian Blaga" University
Department of Informatics
ROMANIA
E-mail: dana.simian@ulbsibiu.ro

CORINA SIMIAN
Zütich University
Institut for Mathematics
SWITZERLAND
E-mail: corina.simian@math.uzh.ch